View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Jagiellonian Univeristy Repository

Machine Dynamics Research
2015, Vol. 39, No 1, 37 - 45

An Outline of a Method of Evaluating the Design Solutions

Andrzej Łachwa
Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian University,
andrzej.lachwa@uj.edu.pl

Abstract

This paper presents an easy to use method to evaluate early design solutions created by a designer and/or generated by a machine. The method is based on the internal representation of design drawings in the form of hierarchical hypergraphs and on the project specification in the form of a list composed of two types of clauses: the elementary ones and the repeatable ones. These clauses represent goals and limitations imposed on the design by the customer. The evaluation value depends on the soft and hard elements of the specification, which are compared with appropriate accurate values of hypergraph attributes.

Keywords: hierarchical hypergraph, graph-based data structure, building design, design knowledge, CAD, design specification, automated compliance checking

1. Introduction

The proposed research aims to develop methods and techniques that support the automated checking of digital building models for compliance with design specification. Essential for automated compliance checking are formal representations of the selected project features and of the specification criteria.

Early design solutions in the form of drawings can be internally represented as attributed hierarchical hypergraphs described in [Grabska et al. 2006, 2009, 2011, 2012, Saaty, 1977].

Design solutions are based on the knowledge of the designer (or the machine) and the specification of the customer. Usually, the goals and limitations that are indicated in the specification are inaccurate and sometimes there are even some discrepancies between them. A project that fits such a specification constitutes a compromise between those goals and limits, as well as the limitations arising from the building code or the regulations of the construction law [Bittermann,2010, Macit et al., 2013].

In this paper we propose a simple formal representation of the design specification and the method of calculating the evaluation of early design solutions.

It will be computed as weighted sum of compatibility degrees between attribute values occurring in formal representations of the design and the specification.

2. Representation of design solutions

In this paper, the internal representation of design drawings in the form of attributed hierarchical hypergraphs has been adopted. The considered hypergraphs are composed of hyperedges corresponding to drawing components (e.g. rooms) and nodes corresponding to fragments of these components (e.g. doors, walls, windows). Hypergraph arcs connecting nodes represent relations among component fragments. Hyperedges represent objects on different levels of detail. Each hyperedge representing an object component can contain a hierarchical hypergraph representing the layout of subcomponents of this component. Nodes and edges in the hypergraph are labelled. The features of the components (or their fragments) selected by the customer and/or designer are represented in the hypergraph by attributes (e.g. surface area, width, type of material) assigned to those elements.

Every project that has been drawn in a relevant system is automatically transformed into the internal representation in the form of an attributed hierarchical hypergraph and specific values are assigned to its attributes. However, some attributes and the methods of calculating their values must be introduced by the designer.

Fig. 1 gives an example of a floor-layout of a small office for two lawyers drawn in a CAD system.

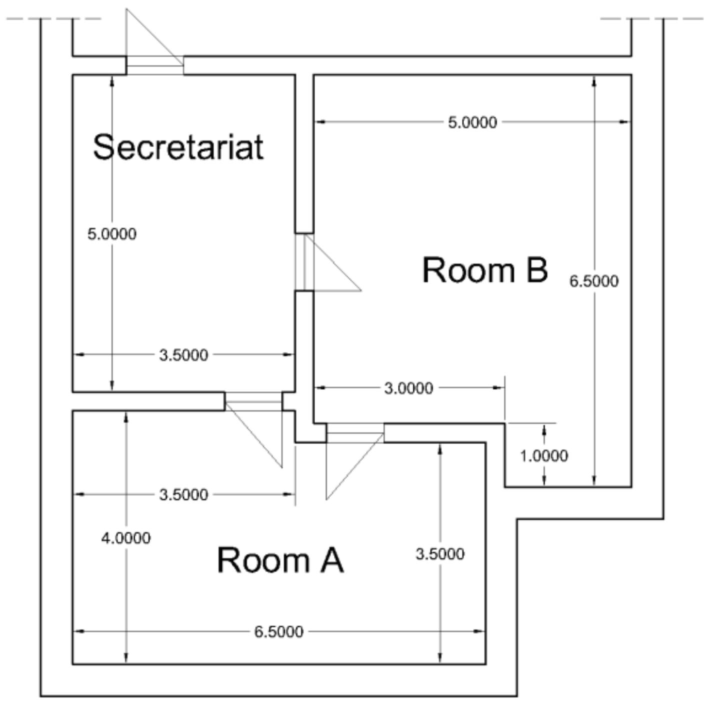

Fig. 1 A floor-layout of the small lawyer's office

The internal representation of the design drawing has the form of hierarchical hypergraph, This kind of graphs has hyperedges, their nodes and arcs linking nodes.

The formal definition of the hypergraph, that has been mentioned here, is as follows [cf. 4]. Let Σ be a fixed alphabet of labels and let Ω be a set of attributes. An attributed hierarchical hypergraph over Σ and Ω is a system $G = (E, V, t, A, lb, att, ch)$, where:

1. E is a nonempty finite set of hyperedges representing object components,
2. V is a nonempty finite set of nodes representing fragments of object components,
3. t: $E \rightarrow V^*$ is a mapping assigning sequences of different nodes to hyperedges,
4. $A \subseteq V \times V$ and $\forall a = (v_1, v_2) \in A\ \exists\, e_1, e_2 \in E$: $e_1 \neq e_2$, $v_1 \in t(e_1)$, $v_2 \in t(e_2)$, is a finite set of arcs representing relations between fragments of components,
5. lb: $E \cup V \cup A \rightarrow \Sigma$ is a labelling function of hypergraph elements,
6. att: $E \cup V \rightarrow 2^{\Omega}$ is an attributing function, where 2^{Ω} is a set of all subsets of Ω,
7. ch: $E \rightarrow 2^{E \cup V \cup A}$ is a child nesting function, such that none hypergraph element can be nested in two different hyperedges, a hyperedge cannot be its own child, and nodes assigned to a nested hyperedge e of E are nested in the same hyperedge as e.

Early design solution in the form of drawings presented in Fig. 1 can be internally represented as attributed hierarchical hypergraphs in a relevant system. The internal representation of the floor-layout of the lawyer's office is shown in Fig. 2. The hypergraph contains four hyperedges (*Secretariat*, *Room A*, *Room B*, *Lawyer's Rooms*), twelve nodes represent the walls and three arcs represent the relation of accessibility. For clarity reasons, in this figure labels of nodes, labels of arcs and attributes are omitted.

The hierarchical hypergraph represents the topology of two office levels and relations between components embedded in hyperedge *Lawyer's Rooms*.

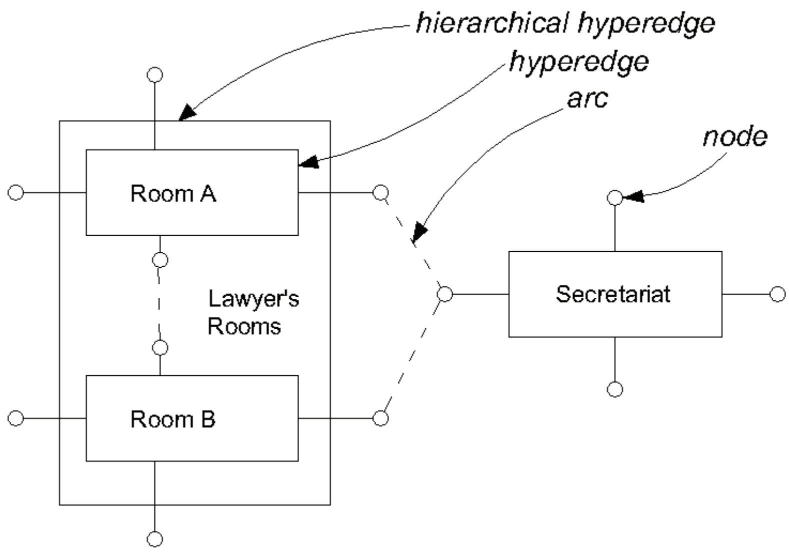

Fig. 2 A hypergraph representation of a floor-layout

The possibility of visualization of relations between different levels during the design process and representing them in the hierarchical hypergraph gives, for example, the prototype CAD system called Hypergraph System Supporting Design and Reasoning (HSSDR) and described in [Grabska et al. 2006]. The HSSDR can also verifies and validates the design solution [Grabska et al. 2011, 2016, Palacz et al., 2011].

3. Representing the specification of the project

In the specification of a project there are goals and limitations imposed on the designer by the customer. Some of these goals and limitations are absolutely required, some are not. They can be inaccurate, impossible to be fulfilled, or even have discrepancies. In such situations, reviewing the initial project conceptions with the customer, as well as analyzing their compliance with the specification, should enable introducing corrections to the specification in a way that will allow covering a greater extent of the goals and limitations of the customer in future designs.

As the model for a specification, we offer here a list of assignments 'attribute ← value, weight', where an attribute means a measurable property of the project, while weight stands for the degree of importance of a particular condition for the investor.

We will divide the assignment clauses into elementary and repeatable. The first type of assignments will stand for such criteria of the specification that concerns a single object or feature (e.g. 'the building must have an underground car park'). The assignment of the second type covers several elements of the same type (e.g. 'the door may not have a span smaller than 90 cm').

Weights should have values in a range [0, 1], where weight 1 stands for a condition absolutely required by the investor. Determination of such weights by the investor, together with the designer, may seem difficult, but there are methods that allow to do this at the relatively low cost. The Saaty method [Saaty, 1977] may serve as an example. In this method, the decision-maker determines the relation between various pairs of criteria by using predicates chosen from a set of several elements (e.g. 'are equally important', 'is slightly more important', 'is much more important'). On the basis of such answers, a matrix is created by substituting the predicates by numbers 1, 3, 5, 7, 9 and their reciprocals. The eigenvector of the matrix that constitutes the maximum value and is properly normalized, is the demanded group of weights that correspond to the evaluation of importance of particular criteria.

The list of assignments represents the investor's specification must be created by the designer.

4. Assessment method

The idea behind the method of automated calculation of project compliance with the specification is to compare relevant pairs of values that occur in two formal models: in the model of the project and in the model of the specification. According to this, the attributes mentioned in the specification must occur in the formal model of the project, that is in the attributed hierarchical hypergraph.

In order to explain the idea of this method, one may, as an example, consider a fragment of a specification for an office building, in which absolutely required conditions occur:

- *number of tiers ← 3, weight 1.00*
- *entrance to the building ← facing the street, weight 1.00*

conditions with a lower weight value, precisely expressed:

- *number of office rooms ← 16, weight 0.80*
- *functions of the ground floor ← reception and exhibition, weight 0.90*

and light conditions:

- *floor plan of the building ← nearing to the square, weight 0.60*
- *total floor space of the building ← about 1200 m2, weight 0.90*
- *shape of the office room ← nearing to the rectangle in ratio 2:3, weight 0.75*
- *floor space of the office room ← about 20 m2, weight 0.80*

The last two conditions determine the properties of the repeatable project element: the office room.

As it already has been mentioned, the attributes that occur in the specification must be present in the representations of the design drawings. The representation of the project in the form of an attributed hierarchical hypergraph allows an open insertion of any attributes of any design component. Thus, an exemplary project created for our specification will have a hypergraph representation, where each edge of the hypergraph that represent the office room will have attributes 'floor space' and 'shape'. It must be highlighted that the attributes in the project representation will take only sharp values, e.g. the shape of the office room will have an assigned numeric value of degree of compliance related to the rectangle shape in ratio of 2:3.

It can be assumed that one of the initial versions of the project of the office building, which the exemplary specification values mentioned above are related to, includes such values as:

- *number of tiers = 2*
- *entrance to the building = facing the street*
- *shape of the floor plan = 0.31 square*
- *number of office rooms = 18*
- *shape of office no. 1 = 0.77 rectangle 2:3*
- *floor space of office no. 1 = 24.5 m2*
- *shape of office no. 2 = 0.38 rectangle 2:3*
- *floor space of office no. 2 = 29.5 m2*

This design does not meet the fixed requirement that the building must have 3 tiers. Other requirements are met on various levels. In particular, the floor spaces of offices no.1 and no.2 exceed the floor space demanded in the specification ('about 20 m2'); the first one exceeds it slightly, the second one – significantly.

The compliance of the whole design with the customer specification will be calculated as an evaluated total of degrees of compliance based on two values of each attribute: the value present in the specification and the value present in the project. In the case of repeatable assignments, the evaluated total shall include the average degree of compliance for all elements of the same type in the project.

The average degree of compliance for the repeatable elements will be calculated as an average quadratic mean of compliance degrees. It can be assumed that among 10 office rooms 9 of them have 24.5 m2 of floor space while one of them has 29.5 m2. Compliance of floor space of those 9 rooms with the specification is significant (0.92), but one room is surely too large: the degree of compliance is only 0.08 (see Fig. 3). The arithmetic mean of compliance degrees is only 0.84, while the quadratic mean is over 0.87. We choose this second average, as the closer to the intuitive evaluation: if 9 rooms meet the requirement of the specification at degree 0.92, then 10 rooms meet this requirement at a slightly lower, but similar, degree, even when the last room is too large.

Fig. 3 explains the way the soft term 'about 20 m2', which is present in the specification, should be understood in. The compliance degree of the floor space of a room with such term is calculated here as the value of function that represents the soft term 'about 20 m2' for the number that constitutes the floor space of a room from the project of the building (the number assigned to the relevant attribute of the hypergraph). Areas of the offices no. 1 and no. 2 correspond to the specification clause 'about 20 m2' in degree 0.92. and in degree 0.08. This way of understanding the soft specification requirements refers to the fuzzy sets theory and their membership functions [Zadeh, 1965, Łachwa, 2001, Łachwa et al., 2006].

In many cases, the linguistic variable technique will also be adopted [Zadeh, 1975, Łachwa, 2001].

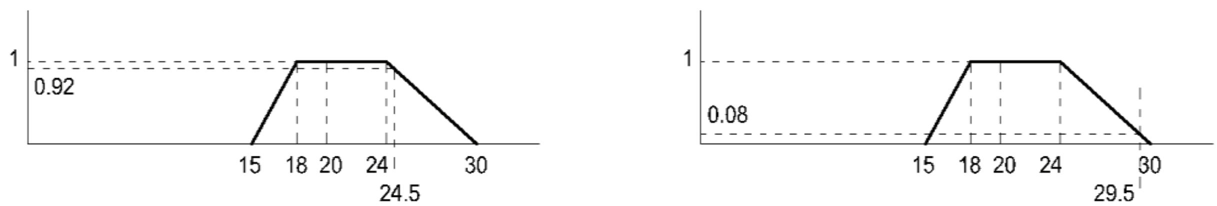

Fig. 3 Compatibility degrees with area about 20 m2

The method of calculating the compliance of the shape proposed in the project with the shape required in the specification should also be explained. To start with, a general measure of shape compliance will be introduced. We will assume that the compliance degree of shape *A* with shape *B* constitutes complements number 1 of quotient of floor space *B* \ *A* and floor space *B*. If required, this quotient will be multiplied by a rate chosen experimentally. According to this proposition, we will define the terms of squareness and rectangleness which occur in this exemplary specification.

Fig. 4 Squereness of two shapes

The degree of squareness of a shape *A* can be computed as max{0, (1–2.3·*a*/*s*)}, where *s* is the area of the smallest square enclosing *A* and *a* is the area of the complement of *A* to *s*. Two examples of shapes are shown in Fig. 4. The left one is close to square in degree 0.80, while the right one only in degree 0.31.

In the same way, the degree of *rectangleness in a ratio 2:3* of a shape *A* can be computed as max{0, (1–1.8·*a*/*r*)}, where *r* is the area of the smallest rectangle in ratio 2:3 enclosing *A* and *a* is the area of the complement of *A* to *r*. In Fig. 5 there are shapes of the offices no. 1 and no. 2. The first shape corresponds to the specification clause '*nearing to the rectangle in ratio 2:3*' in degree (deg_1) 0.77, the second one in degree 0.38. Degrees deg_2 refer to the discussed compliance of those rooms with the requirement that the floor space of each room should be about 20 m2.

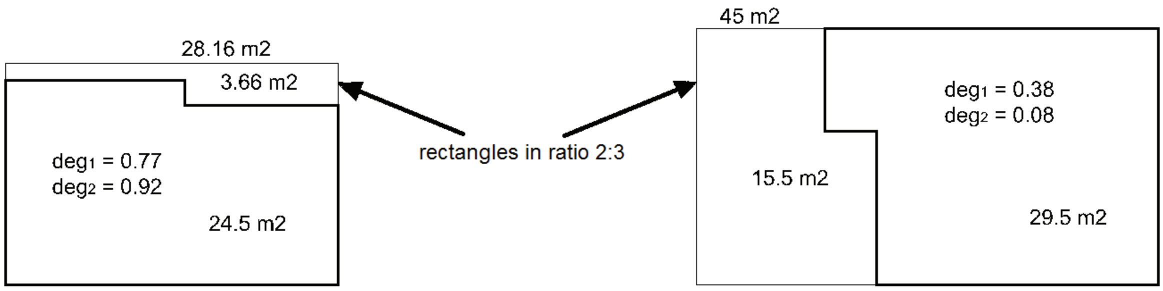

Fig. 5 Two offices

The factors 2.3 and 1.8 in the formulas defining squareness and rectangleness are necessary to ensure the compliance with the natural meaning of the word ‘square’ and of the phrase 'rectangle in ratio 2:3'.

5. Conclusion

The described method of architectural project evaluation refers to their compliance with the specification. However, the full assessment of the project does not constitute a compliance only with the clearly stated goals and requirements of the investor, but also with the evaluation of the esthetic, functional and economic advantages of the design. Even the best project, in terms of its compliance with the specification, will not be completed until it meet the esthetic expectations of the investor, or a limitation concerning a set budget for the investment. These aspects of evaluation have not been discussed in this work.

This paper, as well as the works quoted above [Grabska et al. 2006, 2009, 2011, 2016, Palacz et al.2011] which concern the representation of particular project aspects in a hypergraph, refer only to architectural projects. Nevertheless, it seems that the proposed methods can be successfully used in other areas of designs.

References

Bittermann M. A computational design system with cognitive features based on multi-objective evolutionary search with fuzzy information processing. In: J.S. Gero, ed., *Design Computing and Cognition'10*, 505-524. Kluwer Academic Publishers, 2010.

Grabska E., Borkowski A., Palacz W., Gajek S. *Hypergraph System Supporting Design and Reasoning*, Wolfgang Huhnt (ed.), *Computing in Engineering EG-ICE Conference 2009*, Shaker Verlag, pp. 134-141, 2009.

Grabska E., Grzesiak-Kopeć K., Lembas J., Łachwa A., Ślusarczyk G. Hypergraphs in diagrammatic design. In: K. Wojciechowski et al., eds., *Computational Imaging and Vision*, vol. 32, pp.111-117. Springer, 2006.

Grabska E., Łachwa A., Ślusarczyk G. New visual languages supporting design of multi-storey Buildings. *Advanced Engineering Informatics*, 2012, **26**, pp. 681-690.

Grabska E., Łachwa A., Ślusarczyk G. Approximate Reasoning and Fuzzy Evaluation in Code Compliance Checking. In: A. Gruca et al., eds., Man-Machine Interactions 4, *Advances in Intelligent Systems and Computing* 391, Springer, 2016.

Grabska E., Ślusarczyk G. Knowledge and reasoning in design systems. *Automation in Construction*, 2011, **22**, pp. 927- 934.

Łachwa A. Fuzzy world of sets, numbers, relations, facto, roles and decisions. AOW EXIT, Warszawa 2001 (in Polish).

Łachwa A., Grabska E., Ślusarczyk G. Fuzzy interpretation of layout hypergraphs. *Schedae Informaticae*, 2006, **15**, pp.165-173.

Macit S., İlal M.E., Günaydın H.M., Suter G. İzmir municipality housing and zoning code analysis and representation for compliance checking. In: G. Suter et al., eds., *20th Workshop of the European Group for Intelligent Computing in Engineering*. Vienna University of Technology, 2013.

Palacz W., Grabska E., Gajek S. Conceptual Designing Supported by Automated Checking of Design Requirements and Constraints. D. D. Frey et al. (eds.), Improving Complex Systems Today, *Advanced Concurrent Engineering*, Springer-Verlag, 2011, pp. 257-265.

Saaty T.L. A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 1977, **15**, pp. 234-281.

Ślusarczyk G. Hierarchical hypergraph transformations in engineering design. *Journal Application Computer Science*. 2002, **11**, pp. 67-82.

Zadeh, L.A. Fuzzy sets. *Information and Control*, 1965, **8**: 338–353.

Zadeh, L.A. The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, 1975, **8**, pp. 199–249, 301-357.